

1. Suponga que ejecutamos el siguiente código en C:

```
1 int iter = 0;
2
3 int main() {
4     while(iter < 2){
5         iter++;
6         if(fork()) {
7             printf("Hollow Knight\n");
8             if(iter > 1){
9                 pid_t tt = fork();
10                if(tt > 0){
11                    printf("Ender Lilies\n");
12                }
13                printf("Ender Magnolia\n");
14                exit(1);
15            }
16        }
17    }
18    printf("Silksong\n");
19 }
```

- (a) **(15 puntos)** ¿Cuántos procesos se crean durante la ejecución del código anterior? Explique (puede ser con palabras o con el árbol de procesos que es creado)
- (b) **(15 puntos)** Indique cuántas veces se imprime cada uno de los *strings* del código.
- (c) **(15 puntos)** Explique brevemente (3 líneas como máximo) la razón de las distintas salidas, en términos de orden.

2. Dado los siguientes códigos en C, en donde se ejecuta el programa *programa1*:

```
1 int main() { //programa1
2     pid_t t = fork();
3     if (t > 0){
4         sleep(6);
5         exit(0);
6     }
7     if (t == 0){
8         execlp("./programa2", "", NULL);
9         fork();
10    }
11    fork();
12 }
```

```
1 int main() { //programa2
2     printf("Wooooooooohohoho\n");
3     exit(1);
4     execlp("./programa1", "", NULL);
5     fork();
6 }
```

(15 puntos) Indique la cantidad de procesos que son creados. Explique.